MLPR Lab 3

We will perform image compression on the given image using Principal Component Analysis (PCA). **Do not use** inbuilt python library (sklearn.decomposition.PCA()) to perform PCA. Instead use NumPy functions to calculate PCA.

**Instructions:**

**Step 1:** Import libraries

• OpenCV

• Matplotlib

• NumPy

**Step 2:** Load the given image and read it using OpenCV. Visualise the image.

**Step 3:** Convert the image to grayscale. Visualise it.

**Step 4:** Convert the image to double***(np.float64)*** for performing the mathematical operations accurately.

**Step 5:** Compute the mean of each column (name it ***mean\_column)*** and subtract it from the image (call the subtracted image ***image\_mean\_subtracted)***.

• Visualise this image

**Step 6:** Compute the covariance matrix using numpy on ***image\_mean\_subtracted***.

**Step 7:** Get eigenvalues and eigenvectors using numpy.

**Step 8:** Sort eigenvectors by eigenvalues.

**Step 9:** Define the number of principal components to keep.

***• Num\_components = [10,20,30,40,50,60,90]***, Adjust it check the variations

**Step 10:** for each num\_components, compress the image and then reconstruct it. Store all reconstructed images in ***Output\_images*** variable***.***

***Output\_images = [ ]***

for each ***num\_components***

• Take N number of components and extract eigenvectors.

• Project the data onto the selected components.

***• compressed\_data = image\_mean\_subtracted \* eigen\_matrix***

• Reconstruct the image.

***• reconstructed\_image = compressed\_data \* eigen\_matrix + mean\_column***

• Add the reconstructed image to the list.

• Append reconstructed images to the ***Output\_images[ ]***

**Step 11:** Display the results.

• Provide a common title for all images as “Dimensionality Reduction using PCA.”

• For each image in ***Output\_images,*** display number of components and the explained variance above each plot.

**Step 12**: Now compute minimum num\_components needed to explain 95% variance in data.

• Use the computed eigenvalues to get num\_components

• Now use PCA function from sklearn to compute num\_components (for 95% data variance)

**Submission Instructions:**

• Make sure to add titles and label the axes(where necessary) for all the plots.

• Submit PDF of the code + outputs.

• Add link to the notebook (if using colab). Else upload the code/notebook itself.